Real-time Operating Systems and Systems Programming

Arduino
Lecture 13

ARVUTITEADUSE INSTITUUT
DEPARTMENT OF COMPUTER SCIENCE

"Start with a quote"

1) everything that's already in the world when you're born is just normal;

2) anything that gets invented between then and before you turn thirty is incredibly exciting and creative and with any luck you can make a career out of it;

3) anything that gets invented after you're thirty is against the natural order of things and the beginning of the end of civilisation as we know it until it's been around for about ten years when it gradually turns out to be alright really.

- Douglas Adams

(http://www.douglasadams.com/dna/19990901-00-a.html)

Today

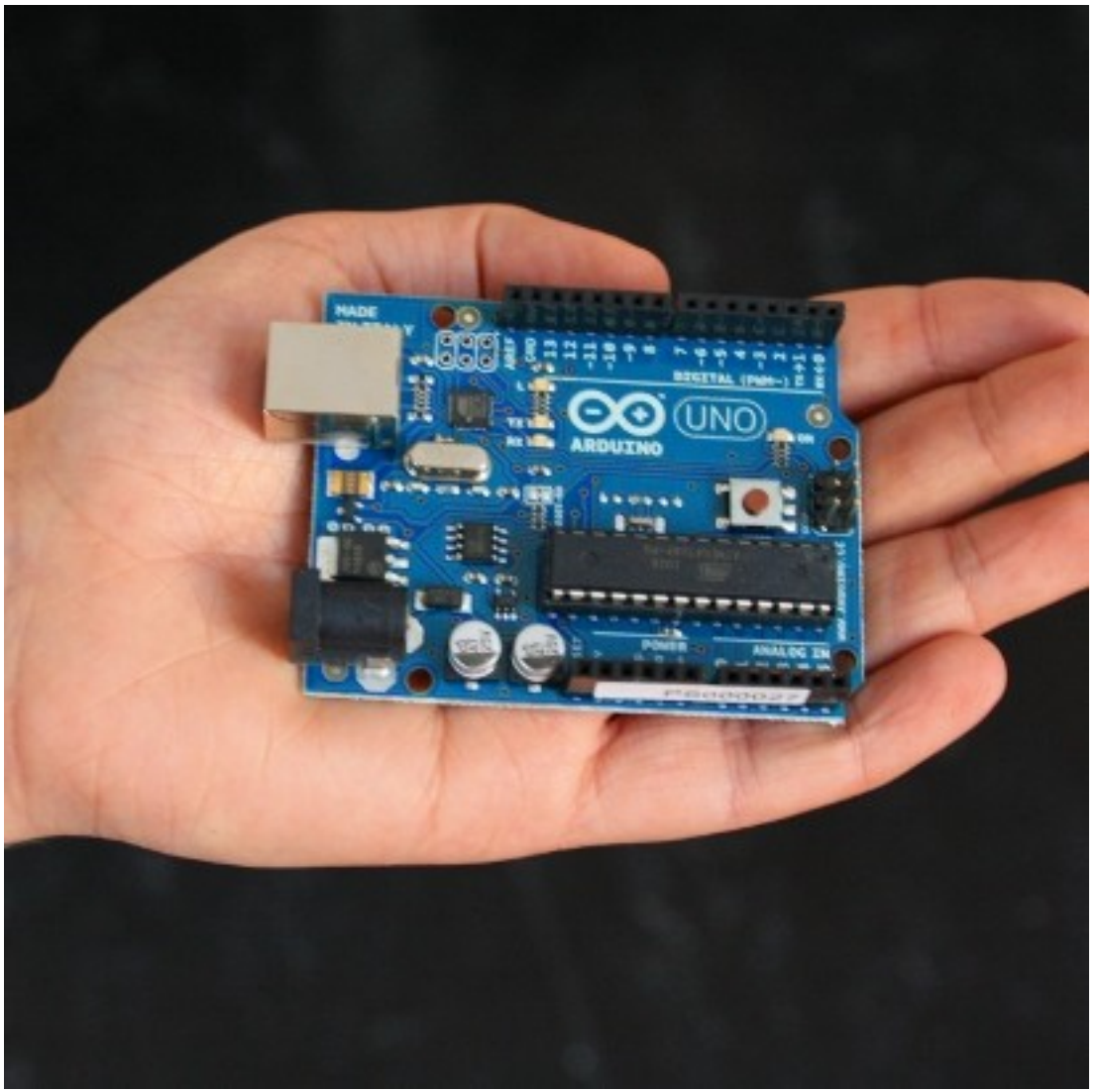
- Arduino demos
- Project background
- Hardware
- Software

What is Arduino

- Arduino is a tool for making computers that can sense and control more of the physical world than your desktop computer.
- It's an **open-source** physical **computing platform** based on a simple microcontroller board, **and** a **development environment** for writing software for the board.

- source: Arduino web, emphasis added

Example: Arduino Uno



Philosophy

- Free as in speech
  - Creative Commons license for layout
  - GPL for software environment
- Designed in Italy, Ivera, to provide students with a cheaper board for easy prototyping.
- Fork of Open source Wiring platform (designed by an artist and programmer Hernando Barragán as his masters thesis)

Hardware

- Schematics and part lists available
  - Create your own
- Also available pre-assembled
  - Under 50€ even then
- Basically Atmel Atmega chip + pins for easy access; USB to serial interface for programming/computer access

Processing

- Open source language interpreter for graphics artists. http://processing.org/
- Designed for non-programmers to start with gratification of visual feedback
- Simplified Java
- *Note to self: Example from Wiki*

Wiring

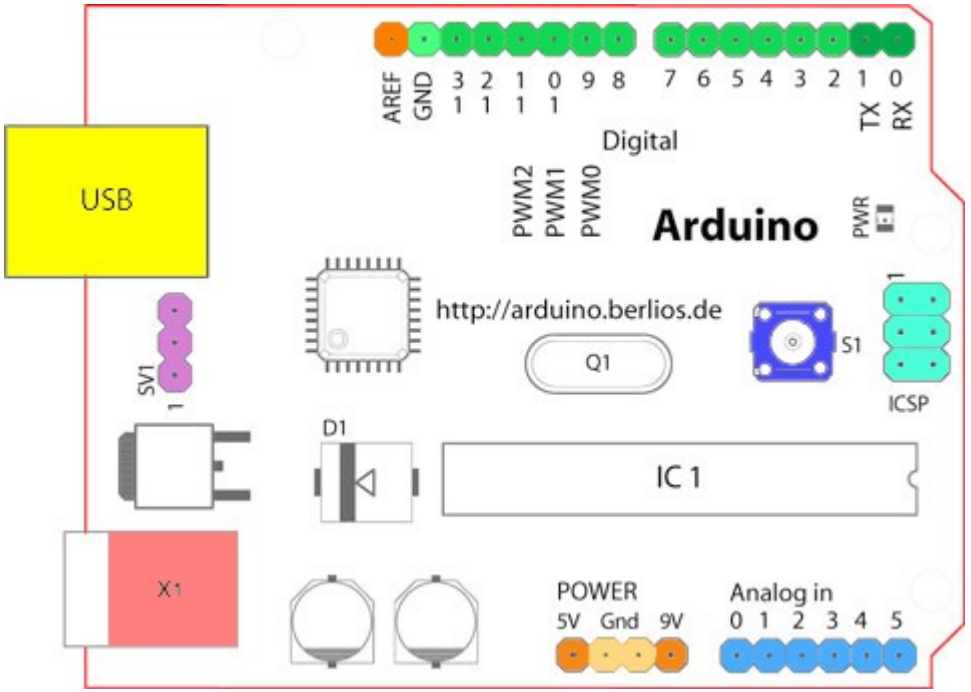
- Used by Arduino (Wiring HW platform also)
- Simplified C/C++
- Defines a header and API
- Programmer writes two functions:
  - setup()
  - loop()
- Compiler adds simple main() and some relevant procedures

Typical First Program

```
#define LED_PIN 13

void setup () {
    pinMode (LED_PIN, OUTPUT);    // enable pin 13 for
digital output
}

void loop () {
    digitalWrite (LED_PIN, HIGH);  // turn on the LED
    delay (1000);                  // wait one second
(1000 milliseconds)
    digitalWrite (LED_PIN, LOW);   // turn off the LED
    delay (1000);                  // wait one second
}
```



Example (from top center)

Reference image (CC) by Arduino

- **Analog Reference** pin (orange)
- **Digital Ground** (light green)
- **Digital Pins** 2-13 (green)
- **Digital Pins** 0-1/**Serial** In/Out - TX/RX (dark green) - These pins cannot be used for digital i/o (digitalRead and digitalWrite) if you are also using serial communication (e.g. Serial.begin).
- **Reset** Button - S1 (dark blue)
- In-circuit **Serial Programmer** (blue-green)
- **Analog In** Pins 0-5 (light blue)
- **Power and Ground** Pins (power: orange, grounds: light orange)
- **External Power** Supply In (9-12VDC) - X1 (pink)
- **Toggles External Power** and USB Power (place jumper on two pins closest to desired supply) - SV1 (purple)
- **USB** (used for uploading sketches to the board and for serial communication between the board and the computer; can be used to power the board) (yellow)

Digital Pins

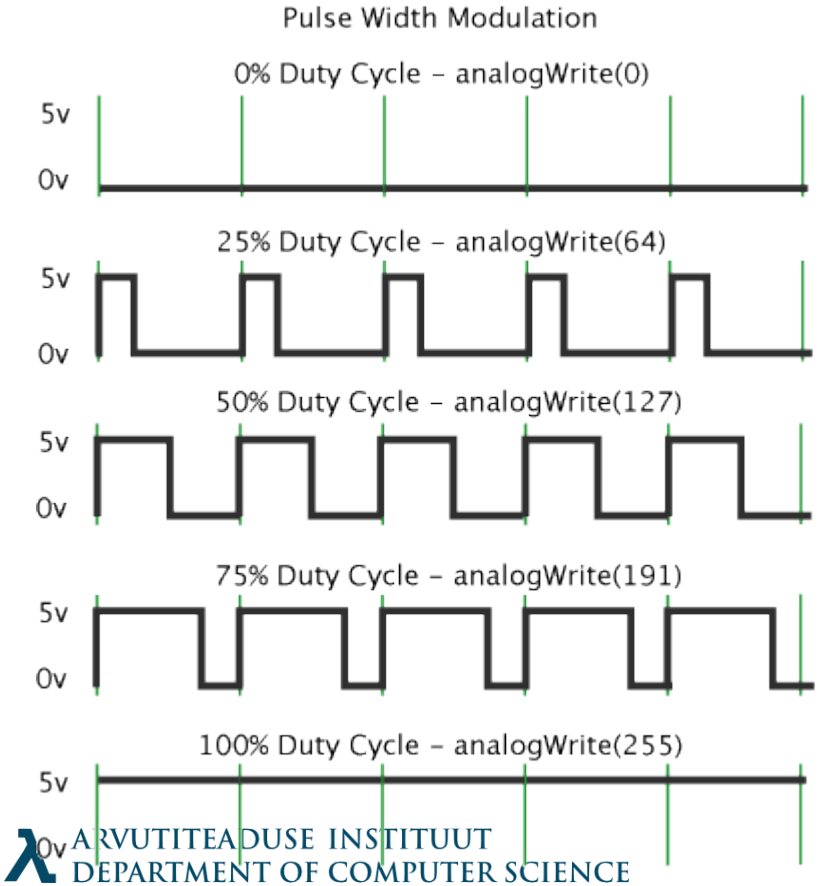
- Serial (0,1)
- External interrupts (2,3)
- PWM (3,5,6,9,10,11)
- SPI (10-14) (not implemented in language)

Analog Pins

- Analog to Digital converter
  - 0-1023
  - val = analogRead(analogPin); // read the input pin
  - From 0V to 5V
- Can be used as Digital output

PWM

- analogWrite(ledPin, val / 4); // analogRead values go from 0 to 1023, analogWrite values from 0 to 255



Memory

- Flash 16k bytes (of which 2k is used for the bootloader)
- SRAM 1024 bytes
  - char message[] = "24 bytes used from RAM.";
- EEPROM 512 bytes
  - EEPROM.read(address);

Interrupts

- Built in handler (on specific pins)
  - attachInterrupt(0, blink, CHANGE); // 0 = intr. num
    void blink() { state = !state; }
- Can be disabled
  - noInterrupts();
  - Interrupts()

Shields

- Plug-in boards which provide additional functionality
- http://shieldlist.org/

Development

- Connect Arduino
- Install Drivers
- Run & configure Arduino IDE
- Create software
- Upload program
- Profit